NoPeek: Information leakage reduction to share activations in distributed deep learning

Praneeth Vepakomma, Abhishek Singh, Otkrist Gupta, Ramesh Raskar
Massachusetts Institute of Technology
Cambridge, MA 02139, USA
{vepakom,abhi24}@mit.edu

2020 International Conference on Data Mining Workshops (ICDMW) | 978-1-7281-9012-9/20/$31.00 ©2020 IEEE | DOI: 10.1109/ICDMW51313.2020.00134

***Abstract*—For distributed machine learning with sensitive data, we demonstrate how minimizing distance correlation between raw data and intermediary representations reduces leakage of sensitive raw data patterns across client communications while maintaining model accuracy. Leakage (measured using distance correlation between input and intermediate representations) is the risk associated with the invertibility of raw data from intermediary representations. This can prevent client entities that hold sensitive data from using distributed deep learning services. We demonstrate that our method is resilient to such reconstruction attacks and is based on reduction of distance correlation between raw data and learned representations during training and inference with image datasets. We prevent such reconstruction of raw data while maintaining information required to sustain good classification accuracies.**

I. Introduction

Data sharing and distributed computation with security, privacy and safety have been identified amongst important current trends in application of data mining and machine learning to healthcare, computer vision, cyber-security, internet of things, distributed systems, data fusion and finance. [1, 2, 3, 4, 5, 2, 6, 7, 8, 9]. Hosting of siloed data by multiple client (device or organizational) entities that do not trust each other due to sensitivity and privacy issues poses to be a barrier for distributed machine learning. This paper proposes a way to mitigate the reconstruction of raw data in such distributed machine learning settings from culpable attackers. Our approach is based on minimizing a statistical dependency measure called distance correlation [10, 11, 12, 13, 14] between raw data and any intermediary communications across the clients or server participating in distributed deep learning. We also ensure our learnt representations help maintain reasonable classification accuracies of the model, thereby making the model useful while also protecting raw sensitive data from reconstruction by an attacker that can be situated in any of the untrusted clients participated in distributed machine learning.

Reconstruction attack setting: The proposed solution aims to give greater protection to the distributed learning ecosystem from reconstruction attacks (reconstruction of raw data from transformed activations) from attackers residing in any client or server that receives communications from another client. It also protects reconstruction attacks from insider threats (attacker resides inside the client/server that transforms the sensitive data). The attack is also illustrated in Figure 1 with regards to the regular training of deep neural networks. We now describe the popular reconstruction attack setting in greater detail along with its relevance to current real-world distributed deep learning prospects.

Attack assumptions: We consider providing security in relatively worst-case settings where the attacker is given an advantage in terms of the assumptions made. This is considered to be a good practice in the community of privacy preserving machine learning as it also enables provision of security under a wider variety of plausible modifications of attack schemes with assumptions that are weaker than the assumed worst-case attacker's capacities. This level of protection is thereby expected to be offered by a working solution in addition to its value in the worst-case setting assumed. In worst-case reconstruction attack settings, the attacker has access to a leaked subset of samples of training data along with corresponding transformed activations at a chosen layer, the outputs of which are always exposed to other clients/server by design for the distributed training of the deep learning network to be possible. The attacker could reside in any untrusted client or server that is part of the distributed training setup. The attacker also has access to rest of the activations corresponding to unleaked training data at the same layer. This is also by design, in order for the distributed training to be functionally possible. The attacker tries to learn an image to image translation model from the transformed activations to the leaked raw data. The attacker can then use that model to reconstruct raw data from activations corresponding to unleaked training data or unleaked test/validation data by inferring from the learnt reconstruction model that was trained on corresponding pairs of activations and raw samples of leaked data.

Attack implications: Typically leakage of a sub-sample of raw data has serious financial, ethical, legal, public relation (PR) and regulatory implications. Such leakages have continued to happen in recent times and often the ratio of $\frac{\text{\# of records leaked}}{\text{total \# of records owned by the institution}}$ is quite small and yet the real-life negative implications of such a leak are massive. According to '2019 Cost of a Data Breach Report' in [15], the cost per data breach is between $\$1.25$ million to $\$8.19$ million depending on country and industry at which the breach occurs. The average size of a data breach is established to be around $25,575$ records. The total global cost of data breaches runs into billions of dollars per year. In addition 60% of small and medium enterprises (SMEs) that experience a cyber breach go out of business in the following 6 months according

2375-9259/20/$31.00 ©2020 IEEE
DOI 10.1109/ICDMW51313.2020.00134

to an official government report in [16]. The lifecycle of a typical data breach is estimated to be 279 days and for that of a malicious attack is estimated to be 314 days. The goal henceforth is to prevent the attacker from using the leaked data to construct an inverse model that can reconstruct other raw data records upon just looking at the communicated intermediate activations received at the server as is required by the important distributed deep learning settings cited above.

Relevance of attack setting: We describe two popular settings of distributed deep learning where this attack setting is highly relevant.

1) **Split Learning:** This attack model is highly relevant to a popular resource and communication efficient variant of distributed deep learning called split learning [17, 18, 19]. In this setting, intermediate activations from a chosen layer (called split leayer) of the deep network are communicated from client to the server during training. The rest of the network is processed at the server during forward propagation. In turn, during backpropagation the gradients from the server's first layer (layer next to the split layer) are communicated back to the client. The rest of backpropagation occurs at the client. These rounds of communication are continued to finish all the epochs of distributed training. Split learning has also been ported into PySyft by OpenMined, a popular and widely adopted opensource framework for privacy preserving distributed machine learning. Split learning has been adopted in Internet of Things (IOT) and edge-device machine learning settings in [20, 21, 22, 23] including multi-modal fusion based machine learning across edge devices in [24, 25] with data collected at the edge on imagery and millimeter wave (mmWave) radio frequency (RF) signals to perform distributed machine learning. Split learning has also been used for Ultra Reliable Low Latency Communication (URLLC) settings with distributed learning as part of 5G communication research. Suitability of split learning for healthcare has been described in [26]. The work in [27] considers protection of intermediate activations under this attack setting solely for private inference via learning of specific noise distributions to perturb the activations prior to communication. Our work instead focuses on training.
2) **Adversarial reconstruction:** This threat model has also been considered with regards to adversarial reconstruction attack settings such as those considered in [28] based on activations obtained at the end of neural network. Server-side insider threats that aim to reconstruct raw data of the client are another realistic example of this attack setting. [29] attempts to learn activations of a given network at chosen layers while attempting to protect an adversarial reconstructor that attempts to reconstruct entire raw data or partial attributes of raw data from these activations.

The attack settings outside the purview of reconstruction attacks that we do not consider in this paper include those of model extraction, model inversion, malicious training, adversarial examples (evasion attacks) and membership inference.

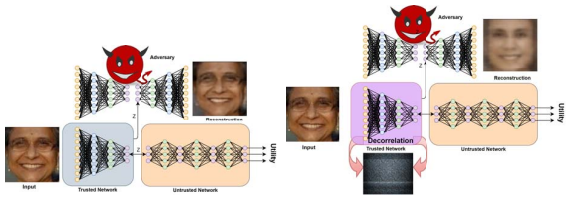

Fig. 1: Left: In the regime of regular training of deep neural networks, information about sensitive raw input data is leaked through intermediate activations even after input data passes through multiple layers. As shown in this figure, upon sending intermediate activations from a trusted network on a client to an untrusted network for computing rest of the task, an adversary on server-side can reconstruct original raw data from the activations. Right: NoPeek is a method where intermediate activations are decorrelated with raw input data while training the network to obtain high classification accuracy on the untrusted network. In this figure, unlike regular deep neural network training, the adversary is not able to reconstruct the exact raw image of the person.

A. Contributions

We show that reducing the distance correlation between learned representations and raw data prevents information leakage with regards to sensitive data in machine learning settings. This is illustrated in Figure 1. The decorrelated data can then be used for various machine learning tasks as long as it holds enough information to perform the intended task while not having enough information to reconstruct the raw data itself. Our developed methods apply to the following settings:

1) Training schemes to prevent reconstruction of entire raw data or specifically chosen attributes in deep learning during inference.
2) Device-level sanitization as burn-in period to reduce leakage of information during the initial epochs of training while not requiring the client that holds the raw data to communicate with the server. Following this burn-in period, the client and server entities train with communication between them.

We evaluate the method and share detailed results via a reconstruction testbed we describe in the experiments section.

B. Code/Reproducibility

The code for our method is provided in this anonymous code repository: https://anonymous.4open.science/r/820473f8-f3ee-4212-9b9c-409a78722af6/. We will also release seeds, trained models, training logs, and intermediate data files for improving reproducibility.

II. Related Work

We primarily focus on the modality of image/computer vision datasets to analyze and test our proposed method. To

maintain specificity we broadly categorized related works on security and privacy for this modality as follows.

a) Deep learning, adversarial learning and information theoretic loss based privacy: These can be categorized into hiding specific sensitive attributes using adversarial training that reaches an equilibrium based on information theoretic loss functions optimized under minimax settings through learning weights of deep learning models [29, 30, 31, 32, 33, 34, 35]. A kernelized version of such an adversarial learning approach with theoretical guarantees is provided in [36]. A similar non-adversarial approach that still uses a dependency measure of maximum mean discrepancy (MMD) for learning a variational autoencoder is in [37, 38]. The method we propose in our paper is not necessarily tied to a generative adversarial network (GAN) styled architecture where two separate models have to be trained in tandem. Our proposed model is based on a easily implementable differentiable loss function between the intermediate activations and the raw data. This will be described in detail later on in section III.

b) Homomorphic encryption and secure multi-party computation for computer vision:: Homomorphic encryption (HE) and multi-party computation (MPC) techniqes although highly secure are not computationally scalable and communication efficient for complex tasks like training large deep learning models. Thereby their application to machine learning has been with regards to smaller computations or specific applications requiring computation of functions with a much simpler complexity. The work in [39, 40, 41] describes HE and garbled circuits (an MPC scheme) for privacy-preserving biometric identification. Similarly [42, 43] uses MPC schemes like oblivious transfer, secure millionaire, secure dot product for face matching with respect to a collection of sensitive surveillance footage images. [44] uses HE for secure aggregation of classifiers in a distributed learning setting where different entities train models on their local data and share the model weights or model related information that needs to be securely aggregated at a centralized entity. Our proposed method in this paper is communication efficient and highly scalable computationally with regards to large deep learning architectures for both training as well as inference attacks unlike the HE and MPC based models.

c) Differential privacy for computer vision:: Differential privacy schemes are based on adding noise dependent on the query under computation to primarily provide privacy against membership inference attacks. The works in [45, 46, 47] are examples of such schemes for transfer learning, subspace clustering. A modified scheme called separated-DP [48] aims to provide guarantees against reconstruction in the context of federated learning [49], a popular distributed learning method for the purpose of protecting model weights trained at individual client entities while securely aggregating the average of these weights at a centralized server. These methods typically take a stronger hit on accuracy of deep learning models although at the benefit of attempting to provide worst-case privacy guarantees for membership inference attacks.

III. Method

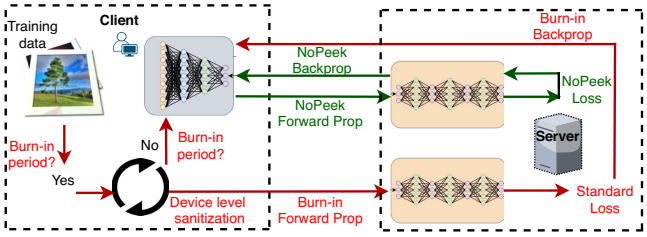

Fig. 2: Workflow for NoPeek method including a) device-level santization on the client during a burn-in period (in red) followed by b) training with the NoPeek method (in green) to prevent reconstruction of data while maintaining classifcation accuracies. There is no communication between client (left) and server (right) during the burning period until the device-level sanitization is completed.

Key idea: The key idea of our proposed method is to reduce information leakage by adding an additional loss term to the commonly used classification loss term of categorical cross-entropy. The information leakage reduction loss term we use is distance correlation; a powerful measure of non-linear (and linear) statistical dependence between random variables. The distance correlation loss is minimized between raw input data and the output of any particularly chosen layer whose outputs need to be communicated from the client to another untrusted client or untrusted server. Optimization of this combination of two losses helps ensure the activations resulting from the protected layer have minimal information with regards to reconstructing the raw data while still being useful enough to achieve reasonable classification accuracies upon post-processing of these activations. The quality of preventing reconstruction of raw input data while maintaining reasonable

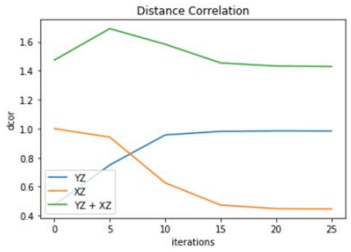

Fig. 3: Universal decorrelation iterations show reduction in distance correlation with raw data (orange) while preserving distance correlation needed to complete the task (blue) on CIFAR-10 data. This scheme is useful to reduce leakage as burn-in period prior to starting distributed deep learning as this does not require any communication with the outside network. This scheme is not required during inference any more as decorrelator is trained by then. We observe in our experiments that this is crucial in preventing reconstruction during initial epochs of distributed training post the burn-in period.

classification accuracies is qualitatively and quantitatively substantiated in the experiments section. Therefore, layers from the raw data upto the protected layer act as decorrelation layers that preserve classification utility.

Loss function: The total loss function for n samples of input data $\mathbf{X}$, activations from protected layer $\mathbf{Z}$, true labels $\mathbf{Y}_{true}$, predicted labels $\mathbf{Y}$ and scalar weights α_1, α_2 is therefore given by

$$\alpha_1 DCOR(\mathbf{X}, \mathbf{Z}) + \alpha_2 CCE(\mathbf{Y}_{true}, \mathbf{Y}) \tag{1}$$

The gradient of distance correlation is provided below for optimization purposes in Appendix A although we optimize the above loss function using *Autograd* as we use it in the context of distributed deep learning. A deep learning friendly code for computing distance correlation is also provided there. A link to our anonymous code repository is also provided there for reproducibility.

A. Initialization with device-level decorrelation

All iterations of training require communication between the client/server entities involved in distributed deep learning. In order to ensure that there is no leakage during the initial iterations of minimizing our proposed loss function, we perform a device-level decorrelation routine during an initial burn-in period of iterations before allowing any communication. Therefore this is a highly communication-efficient approach for learning decorrelated representations of client's raw data that can then be shared with other entities in a distributed learning setting. Following the burn-in period, the server receives the decorrelated activations and continues to sync with the client to perform distributed training of our loss function proposed in previous subsection. This process is illustrated at a high-level in Figures 2, 3, 4 and 5.

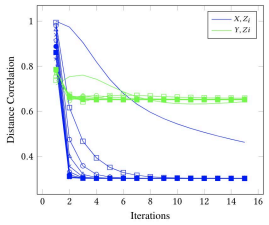

Fig. 4: Universal decorrelation iterations show reduction in distance correlation with raw data (blue) while preserving distance correlation needed to complete the task (green) on Boston Housing data. This scheme is useful to reduce leakage as burn-in period prior to starting distributed deep learning as this does not require any communication with the outside network.

Unlike traditional distributed deep learning approaches, this iterative approach does not require any gradient backpropagation or exchange of activations with outside network to perform the optimization and is particularly suitable for on-device (client) decorrelation of data prior to performing any major communications across the distributed entities. This scheme is useful to reduce leakage as burn-in period prior to starting distributed deep learning as it does not require any communication with the outside network. Following this after the distributed deep learning based decorrelator is trained, this scheme is not required during inference any more as decorrelator is trained to be optimal by that point of time. This approach of solely optimizing on client to initialize the distributed deep learning model with a decorrelated representation prior to beginning model training as illustrated in Figure 5.

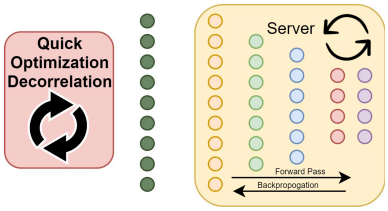

Fig. 5: The device level sanitization scheme helps to control leakage during early iterations of model training by acting as a burn-in period to initialize the activations for the deep learning based decorrelator. Following this the distributed deep learning model is trained. During inference, the distributed deep learning model is used directly as the model remains trained at that point of time.

The idea is based on a modification of a scheme for maximizing sum of distance correlations proposed in [50] to obtain low-dimensional representations that preserve a high distance correlation with labels $\mathbf{Y}$. Motivated by their setup which also seems similar in principle to the information bottleneck method, we instead consider a *difference* of these distance correlations instead of a sum as unlike their method which was for supervised dimensionality reduction we would like to minimize distance correlation with raw data while preserving distance correlation with labels. This objective function can be expressed as below as distance correlation can be expressed in terms of specific graph Laplacians whose exact form is detailed in [50]:

$$f(\mathbf{Z}) = \frac{\operatorname{Tr} \mathbf{Z}^T \mathbf{L_y} \mathbf{Z}}{\sqrt{\operatorname{Tr} \mathbf{Y}^T \mathbf{L_Y} \mathbf{Y} \operatorname{Tr} \mathbf{Z}^T \mathbf{L_Z} \mathbf{Z}}} - \frac{\operatorname{Tr} \mathbf{Z}^T \mathbf{L_X} \mathbf{Z}}{\sqrt{\operatorname{Tr} \mathbf{X}^T \mathbf{L_X} \mathbf{X} \operatorname{Tr} \mathbf{Z}^T \mathbf{L_Z} \mathbf{Z}}}$$

The iterative update for maximization of the above objective is based on a variant of majorization-minimization [51, 52] and is given by $\mathbf{Z}_t = \mathbf{H}\mathbf{Z}_{t-1}$ where

$$\mathbf{H} = \left(\gamma^2 \mathbf{D}_X - \alpha \mathbf{S}_{\mathbf{X},\mathbf{Y}}\right)^{\dagger} \left(\gamma^2 \mathbf{D}_X - \mathbf{L}_{\mathbf{M}}\right)$$

for a fixed γ^2, some α and where $k_X = \frac{1}{\sqrt{\operatorname{Tr} \mathbf{X}^T \mathbf{L_X} \mathbf{X}}}$, $k_Y = \frac{1}{\sqrt{\operatorname{Tr} \mathbf{Y}^T \mathbf{L_Y} \mathbf{Y}}}$ are constants, and $\mathbf{S}_{\mathbf{X},\mathbf{y}} = k_Y \mathbf{L_Y} - \beta k_X \mathbf{L_X}$ for a tuning parameter β, where the details of all these parameters are also in [50] except for β which has been added to study its effect on convergence rate as seen in Figure 2. The only other difference in the iterative updates ends up in the definition of $\mathbf{S_{XY}}$ where in the case of the supervised dimensionality reduction usecase, the negative instead becomes a positive.

B. Advantages of using distance correlation

Estimation of classical information theoretic-measures as used in [30, 31, 32, 33, 34] is a known hard problem. Recent approaches to estimate it effectively like [53] are based on

iterative optimization. A recent data efficient version of it requires 3 nested for loops of optimization [54] to estimate it. Therefore in the context of deep learning, every epoch of learning the weights is dependent on this iterative optimization. In contrast our approach uses distance correlation, a measure of non-linear (and linear) dependency that can be estimated in closed-form. Fast estimators of distance correlation require $\mathcal{O}(nlogn)$ [55, 56] computational complexity for univariate and $\mathcal{O}(nKlogn)$ complexity [57] for multivariate settings with $\mathcal{O}(max(n, K))$ memory requirement, where K is the number of random projections. Distance correlation has been shown to be a simpler special case of other recent popular measures of dependence such as Hilbert-Schmidt Independence Criterion (HSIC), Maximum Mean Discrepancy (MMD) and Kernelized Mutual Information (KMI) that have been extensively studied and used recently in the machine learning and statistics community [11, 58, 59, 60]. An advantage of using a simpler alternative is that in addition to it being differentiable and easily computable with a closed-form, it requires no other tuning of parameters and is self-contained unlike HSIC, MMD and KMI that depend on a choice of separate kernels for features as well as labels along with their respective tuning parameters.

IV. Experiments

Reconstruction attack testbed: We empirically examine the privacy aspects of our method by designing a testbed which performs feature inversion. The idea of this testbed is to emulate the attacker and to evaluate the quality of reconstruction attack. The testbed itself is a neural network with a decoder architecture where the layers are composed of transpose convolutions. Similar architecture have been used in generative models for generating images from low-dimensional latent codes. We use other standard components like ReLU activation, batch normalization and ResNet style of performing additive skip connections. Input to this testbed is the intermediate activations, z_l from any arbitrary layer l of the target model and output is the image generated $\hat{x}$. We first train two separate ResNet-18 architectures on datasets for image classification with NoPeek and without NoPeek for baseline comparison. After the training, we use held-out validation set to generate intermediate activations. We thereby generate a paired dataset of activations and corresponding images. We use this paired dataset to train the reconstruction testbed to emulate the attacker. We use 90% of the original validation dataset as training dataset for the reconstruction testbed and remaining 10% is used as test-set for the qualitative evaluation of the reconstruction quality. We train the model in the reconstruction testbed on a dataset of z_l, x pairs with the loss function as the euclidean norm between x and $\hat{x}$. We want to emphasize that there can be a potentially better design for architectures of the reconstruction testbed and better loss functions but the goal of this paper is to just have a fair comparison between the NoPeek based training and the regular training of deep networks using a reasonable reconstruction architecture. The number of upsampling layers in the architecture of the testbed vary depending upon the difference in the dimensionality of z_l and x.

Fig. 6: Reconstruction results for CIFAR10. The top row is the original image and the second row is reconstruction from activations of the network trained on CIFAR10. In the third row, activations are presented from network trained with NoPeek method. Even though the images are part of the training dataset, the network present in reconstruction is not able to generate any meaningful or discriminative representation when activations come from the network trained with NoPeek. The testbed fails to reconstruct the activations perfectly for the baseline as well especially because the activations belong to last layer.

For all of our reported experiments for training the network, we use Adam optimizer with initial learning rate as $1 \times e^{-3}$ with exponential decay. The experiments are further detailed below.

A. CIFAR10

We use CIFAR10 for our inversion experiments with respect to vision model trained for image classification task. We first train the network on 50,000 training samples of CIFAR10 with and without NoPeek and then use 10,000 validation data samples and their corresponding activation as the dataset for the inversion attack model. We choose one of the middle layers of ResNet-18, that is positioned at second stage and first residual block in the network. We train the ResNet-18 with both regular training (baseline) and NoPeek approaches. We then use this layer's activations on the validation as the input dataset for training the reconstruction testbed. The result is shown in the Figure 6. The low resolution of images in the CIFAR10 dataset makes it difficult to report the results experimentally. However, degradation in the reconstruction quality is clearly evident in the Figure 6.

B. UTKFace

UTKFace is a database of human faces. We train a ResNet based branched network which learns to predict the age, race, and gender of the person. The initial part of the network is common for all three prediction branches and splits near the end of the network with dense layers. We use ResNet-18 for the common part of the architecture and choose the last ResNet block of the output of second stage to be decorrelated with the input for testing the NoPeek approach.

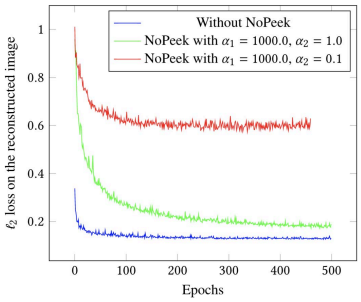

Fig. 7: average ℓ_2 norm between the image reconstructed by reconstruction testbed and original image in UTKFace dataset. Changing α results in different levels of the difference in the . ℓ_2 loss between the two images is not the ideal metric as it does not handle the semantic features present in the image. For the qualitative results please see Figure 8.

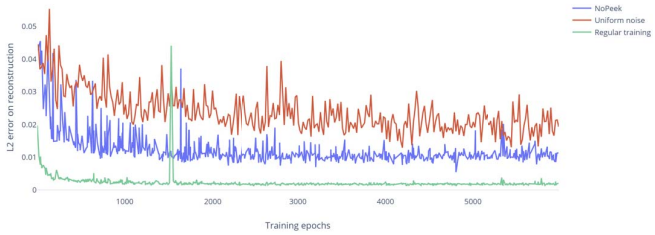

Fig. 8: **Privacy-utility tradeoff on UTK:** We show l_2 error of reconstruction of a baseline strategy of adding uniform noise (in red) to activations of the layer being protected. This results in a model of no classififcation utility (performs at chance accuracy) albeit while preventing reconstruction. Our NoPeek approach (in blue) attains a much greater classfication accuracy for the downstream task (0.82) compared to adding uniform noise (chance accuracy) while still preventing reconstruction of raw data. This is compared to regular training, that does not prevent the reconstruction (in green).

Figure 9 shows the qualitative result of our experiment on the face attribute prediction. We observe majority of the faces to be unidentifiable from the reconstruction.

In general, we observe the reconstructions learned by the testbed trained on NoPeek activations tend to be relatively similar towards average face image of this dataset. For quantitative comparison, we plot the average l_2 reconstruction error for the entire test dataset in Figure 7. In Figure 8. we show the privacy-utility tradeoff with respect to a baseline of adding uniform noise, NoPeek and conventional training.

C. Diabetic Retinopathy

Privacy is a well known concern in the medical community hence, we experiment NoPeek approach for training network on the task of Diabetic retinopathy severity detection method. Previous research [61] has shown it is possible to predict personal attributes of a person like gender, smoking habits etc. from fundus images. In our experiment, we train a CNN

Fig. 9: We use the reconstruction testbed to generate faces from the activations of a given intermediate layer. Here, first row is the actual image, second row is reconstruction from the activations and third row is reconstruction when the network is trained with NoPeek. NoPeek training makes it difficult for the adversary to generate the actual image from the activations.

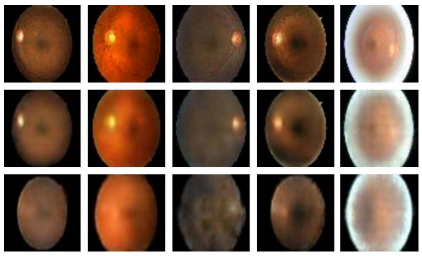

Fig. 10: Reconstruction results for fundus images with first row being the original image, second row as reconstruction from regular training and third is the result of reconstruction from NoPeek. The finer level granularity of vessels and dark spots is lost in both images but the no-peek approach loses it even more making better for privacy of attributes which can be inferred from these finer level details which is needed for diagnosis or biometric applications.

model to predict diabetic retinopathy severity from the fundus images. We use standard ResNet-18 for training the main model and partition the activations in the middle of the layers. We downsample the fundus images to standardize them to a common size of 64×64. Hence, the task for the reconstruction testbed is to generate 64×64 fundus image given the intermediate activations. Figure 10 shows qualitative results for some of the samples, it can be noted that the images generated by the testbed do not reconstruct attribute discriminative features such as blood vessels successfuly and the loss of such discriminative features is higher for the NoPeek method.

D. Attribute Privacy

As described mathematically in the section 4.2, we also extend the NoPeek approach to minimize distance correlation between the intermediate activations and a particular chosen attribute with respect to which we want to attain privacy. We use UTK Face dataset under the same setting as described in the section 7.2, but this time instead of training the method on all three attributes - age, gender, and race, we only train on the two attributes at a time while treating the other attribute as

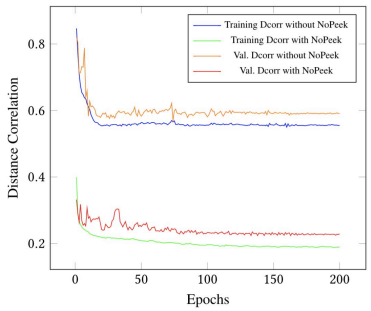

Fig. 11: Distance correlation value as the network gets trained on the UTK face dataset. Note that the network by itself reduces distance correlation even in the baseline experiment.

a protected class and hence, distance correlation is minimized between the intermediate activations and the corresponding attribute. We do not treat gender as a protected attribute since

Fig. 12: We extend the NoPeek approach to decorrelate the activations with respect to a particular attribute. In this figure, we decorrelate the intermediate activations with the attribute (race) we want to protect from leaking during inference. Usually, the information leakage is still very high in the activations upon applying only a single layer of convolution filters, yet, we can see here that the NoPeek approach makes it difficult for the adversary to reconstruct the person's image.

it is a binary class in our dataset. Figure 12 shows the reconstruction results when NoPeek approach is used for attribute level privacy. The more compelling part of this experiment is that the reconstruction testbed is never supervised about the image's attributes itself, it is just designed to generate the image showing that attribute level privacy technique is indeed capturing related attributes and invariances.

E. Visualizing activations

We visualize the activations of the filters in early layers to see the resulting effect of minimizing the distance correlation between activations and raw data. Figure 14 shows the decreasing levels of leakage.

For this experiment, we treat the target z for NoPeek as the output of first CNN layer itself which we restrict to only three output channels so as to visualize only the RGB component. Figure 15 shows the output of first layer of the trained network. The joint minimization of distance correlation with cross entropy(in classification task) leads to a different set of feature extraction or transformation over features in such a way that it is perceivable for human visual system as well.

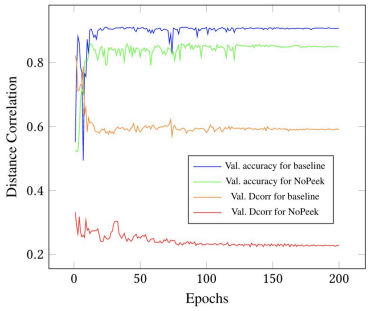

Fig. 13: By introducing NoPeek in the training of the network, we obtain a major decrease in the distance correlation from 0.6 (baseline) to 0.22 (NoPeek) while the decrease in the accuracies is relatively much lesser.

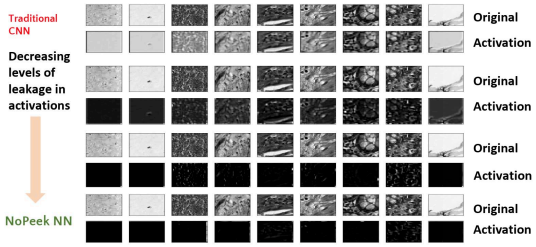

Fig. 14: We see decreasing levels of leakage of information about raw data in the activations as the weight of distance correlation term in the weighted loss function is increased significantly over a colorectal histology medical dataset available publicly.

We also visualize activations obtained on colorectal histology dataset with increasing values of α in 14.

V. Discussion

One of the important aspects of proposed technique is to jointly optimize for distance correlation and task related loss function like cross entropy for classification. In other words, we are optimizing for the trade-off between privacy and utility by controlling α_1 and α_2 as described previously. Figure 7 shows three variations of this trade-off , where *Without NoPeek* approach is essentially $\alpha_1 = 0$. In order to understand it well, in Figure 11 we plot the distance correlation of a fixed intermediate activation as during training for a NoPeek network

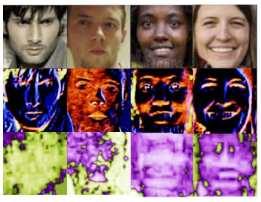

Fig. 15: Visualization of the activations of the first layer of a ResNet. In the activation maps of the first layer in the second row, subtle facial features can be observed from the activations about the raw image while, in the third row, the NoPeek method forces the network to decorrelate the features with respect to raw data, hence making it hard to interpret.

as well as for a network without NoPeek. This demonstrates that the network without NoPeek also reduces the distance correlation beyond a certain layer and our proposed method can be seen as an additional regularizer which forces the network to regularize for the reduction in distance correlation at a much higher rate between raw data and the activations. In Figure 13 we observe that accuracy dropped by a relatively small amount compared to the drop in distance correlation and this relative difference between the drop can be controlled through tuning the α for the distance correlation as well as cross entropy. The choice of α is also dictated by the position of the intermediate activation in the network as well as the type of layer which produces the output. For example, compared to convolution layers, which imposes heavy prior on images, a fully connected layer can learn to attain a lesser distance correlation relatively easy and hence should also guide the choice of α.

VI. Conclusion

The proposed NoPeek schemes based on distance correlation seem to have versatile applicability in the space of privacy, computer vision and machine learning given that it does not require major changes in the model setup and architectures except for the proposed modification to loss function.It would be great to realize on-device implementations of the universal decorrelation and other NoPeek schemes. With regards to human visual perception of bias and privacy, we would also like to conduct a large-scale crowdsourced survey to compare performance of human participants in deciphering the true sensitive attribute upon looking at NoPeek results in terms of their proximity to a uniform random prediction.

Appendix

Distance correlation between centered data can be represented as $\frac{\text{Tr}\,(\mathbf{X^T X Z^T Z})}{\sqrt{\text{Tr}\,(\mathbf{X^T X})^2\,\text{Tr}\,(\mathbf{Z^T Z})^2}}$ [50]. Distance covariance in the numerator can be written as

$$\text{Tr}(\mathbf{X^T Z X}) = \sum_{ij} \langle z_i, z_j \rangle (\|x_i - x_j\|)^2 \tag{2}$$

This can be written in matrix form using basis vectors e_i, e_j as

$$\sum_{ij} [\text{Tr}\,(\mathbf{Z^T e_i e_j^T Z}) \mathbf{Tr}(\mathbf{X^T (e_i - e_j)(e_i - e_j)^T X})] \tag{3}$$

Simplifying the notation with $M_{ij} = e_i e_j^T$ and $A_{ij} = (e_i - e_j)(e_i - e_j)^T$ we have

$$\frac{\partial Tr(\mathbf{Z^T L_Z Z})}{\partial \mathbf{Z}} = \sum_{ij} (\mathbf{2 M_{ij} Z}) \mathbf{Tr}(\mathbf{X^T A_{ij} X})$$

On the lines of 3, we have $Tr(\mathbf{Z^T L_Z Z}) = \sum_{ij} [Tr(\mathbf{Z^T M_{ij} Z}) \mathbf{Tr}(\mathbf{Z^T A_{ij} Z})]$ Therefore utilizing these identities, the derivative of squared distance correlation w.r.t $\mathbf{Z}$ can be written as

$$\frac{c_x Tr(\mathbf{Z^T L_Z Z}) \frac{\partial Tr(\mathbf{X^T L_Z X})}{\partial \mathbf{Z}} - [Tr(\mathbf{X^T L_Z X})]^2 c_x \frac{\partial Tr(\mathbf{Z^T L_Z Z})}{\partial \mathbf{Z}}}{[Tr(\mathbf{Z^T L_Z Z})]^2} \tag{4}$$

A. B: Deep-learning friendly source code for sample distance correlation

```
def pairwise_dist(A):
    r = tf.reduce_sum(A*A, 1)
    r = tf.reshape(r, [-1, 1])
    D = tf.maximum(r - 2*tf.matmul(A, tf.transpose
    D = tf.sqrt(D)
    return D

def dist_corr(X, Y):
    n = tf.cast(tf.shape(X)[0], tf.float32)
    a = pairwise_dist(X)
    b = pairwise_dist(Y)
    A = a - tf.reduce_mean(a, axis=1) -\
    tf.expand_dims(tf.reduce_mean(a,axis=0),axis=1
    tf.reduce_mean(a)
    B = b - tf.reduce_mean(b, axis=1) -\
    tf.expand_dims(tf.reduce_mean(b,axis=0),axis=1
    tf.reduce_mean(b)
    dCovXY = tf.sqrt(tf.reduce_sum(A*B) / (n ** 2)
    dVarXX = tf.sqrt(tf.reduce_sum(A*A) / (n ** 2)
    dVarYY = tf.sqrt(tf.reduce_sum(B*B) / (n ** 2)

    dCorXY = dCovXY / tf.sqrt(dVarXX * dVarYY)
    return dCorXY
```

B. Code/Reproducibility

The code for our method is provided in this anonymous code repository: https://anonymous.4open.science/r/820473f8-f3ee-4212-9b9c-409a78722af6/.

References

[1] P. Aditya, R. Sen, P. Druschel, S. Joon Oh, R. Benenson, M. Fritz, B. Schiele, B. Bhattacharjee, and T. T. Wu, "I-pic: A platform for privacy-compliant image capture," in *Proceedings of the 14th annual international conference on mobile systems, applications, and services*. ACM, 2016, pp. 235–248.

[2] T. Orekondy, B. Schiele, and M. Fritz, "Towards a visual privacy advisor: Understanding and predicting privacy risks in images," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 3686–3695.

[3] J. Schiff, M. Meingast, D. K. Mulligan, S. Sastry, and K. Goldberg, "Respectful cameras: Detecting visual markers in real-time to address privacy concerns," in *Protecting Privacy in Video Surveillance*. Springer, 2009, pp. 65–89.

[4] A. Frome, G. Cheung, A. Abdulkader, M. Zennaro, B. Wu, A. Bissacco, H. Adam, H. Neven, and L. Vincent, "Large-scale privacy protection in google street view," in *2009 IEEE 12th international conference on computer vision*. IEEE, 2009, pp. 2373–2380.

[5] X. Yu and N. Babaguchi, "Privacy preserving: hiding a face in a face," in *Asian Conference on Computer Vision*. Springer, 2007, pp. 651–661.

[6] M. Upmanyu, A. M. Namboodiri, K. Srinathan, and C. Jawahar, "Efficient privacy preserving video surveillance," in *2009 IEEE 12th international conference on computer vision*. IEEE, 2009, pp. 1639–1646.

[7] A. Senior, S. Pankanti, A. Hampapur, L. Brown, Y.-L. Tian, A. Ekin, J. Connell, C. F. Shu, and M. Lu, "Enabling video privacy through computer vision," *IEEE Security & Privacy*, vol. 3, no. 3, pp. 50–57, 2005.

[8] S. Avancha, A. Baxi, and D. Kotz, "Privacy in mobile technology for personal healthcare," *ACM Computing Surveys (CSUR)*, vol. 45, no. 1, p. 3, 2012.

[9] D. Halperin, T. S. Heydt-Benjamin, K. Fu, T. Kohno, and W. H. Maisel, "Security and privacy for implantable medical devices," *IEEE pervasive computing*, vol. 7, no. 1, pp. 30–39, 2008.

[10] G. J. Székely, M. L. Rizzo, N. K. Bakirov *et al.*, "Measuring and testing dependence by correlation of distances," *The annals of statistics*, vol. 35, no. 6, pp. 2769–2794, 2007.

[11] D. Sejdinovic, B. Sriperumbudur, A. Gretton, K. Fukumizu *et al.*, "Equivalence of distance-based and rkhs-based statistics in hypothesis testing," *The Annals of Statistics*, vol. 41, no. 5, pp. 2263–2291, 2013.

[12] G. J. Székely, M. L. Rizzo *et al.*, "Brownian distance covariance," *The annals of applied statistics*, vol. 3, no. 4, pp. 1236–1265, 2009.

[13] G. J. Székely and M. L. Rizzo, "Energy statistics: A class of statistics based on distances," *Journal of statistical planning and inference*, vol. 143, no. 8, pp. 1249–1272, 2013.

[14] J. Dueck, D. Edelmann, T. Gneiting, D. Richards *et al.*, "The affinely invariant distance correlation," *Bernoulli*, vol. 20, no. 4, pp. 2305–2330, 2014.

[15] IBM and P. Institute, "Cost of a data breach report," *IBM White Paper*, 2019.

[16] A. goverment report, "Cyber security best practice research report," *Official Government Report: Australian small business and family enterprise ombudsman*, 2017.

[17] O. Gupta and R. Raskar, "Distributed learning of deep neural network over multiple agents," *Journal of Network and Computer Applications*, vol. 116, pp. 1–8, 2018.

[18] P. Vepakomma, O. Gupta, T. Swedish, and R. Raskar, "Split learning for health: Distributed deep learning without sharing raw patient data," *arXiv preprint arXiv:1812.00564*, 2018.

[19] A. Singh, P. Vepakomma, O. Gupta, and R. Raskar, "Detailed comparison of communication efficiency of split learning and federated learning," *arXiv preprint arXiv:1909.09145*, 2019.

[20] Y. Koda, J. Park, M. Bennis, T. Nishio, K. Yamamoto, M. Morikura, and K. Nakashima, "Communication-efficient multimodal split learning for mmwave received power prediction," *IEEE Communications Letters*, 2020.

[21] Y. Gao, M. Kim, S. Abuadbba, Y. Kim, C. Thapa, K. Kim, S. A. Camtepe, H. Kim, and S. Nepal, "End-to-end evaluation of federated learning and split learning for internet of things," *arXiv preprint arXiv:2003.13376*, 2020.

[22] C. Thapa, M. Chamikara, and S. Camtepe, "Splitfed: When federated learning meets split learning," *arXiv preprint arXiv:2004.12088*, 2020.

[23] W. Y. B. Lim, J. S. Ng, Z. Xiong, D. Niyato, C. Leung, C. Miao, and Q. Yang, "Incentive mechanism design for resource sharing in collaborative edge learning," *arXiv preprint arXiv:2006.00511*, 2020.

[24] Y. Koda, J. Park, M. Bennis, K. Yamamoto, T. Nishio, and M. Morikura, "One pixel image and rf signal based split learning for mmwave received power prediction," in *Proceedings of the 15th International Conference on emerging Networking EXperiments and Technologies*, 2019, pp. 54–56.

[25] J. Park, S. Samarakoon, H. Shiri, M. K. Abdel-Aziz, T. Nishio, A. Elgabli, and M. Bennis, "Extreme urllc: Vision, challenges, and key enablers," *arXiv preprint arXiv:2001.09683*, 2020.

[26] B. Allen, S. Agarwal, J. Kalpathy-Cramer, and K. Dreyer, "Democratizing ai," *Journal of the American College of Radiology*, vol. 16, no. 7, pp. 961–963, 2019.

[27] F. Mireshghallah, M. Taram, P. Ramrakhyani, D. Tullsen, and H. Esmaeilzadeh, "Shredder: Learning noise distributions to protect inference privacy," 2019.

[28] Z. Yang, J. Zhang, E.-C. Chang, and Z. Liang, "Neural network inversion in adversarial setting via background knowledge alignment," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 225–240. [Online]. Available: https://doi.org/10.1145/3319535.3354261

[29] A. Li, J. Guo, H. Yang, and Y. Chen, "Deepobfuscator: Adversarial training framework for privacy-preserving image classification," *arXiv preprint arXiv:1909.04126*, 2019.

[30] V. Mirjalili, S. Raschka, and A. Ross, "Flowsan: Privacy-enhancing semi-adversarial networks to confound arbitrary face-based gender classifiers," *arXiv preprint arXiv:1905.01388*, 2019.

[31] P. C. Roy and V. N. Boddeti, "Mitigating information leakage in image representations: A maximum entropy approach," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2586–2594.

[32] B. H. Zhang, B. Lemoine, and M. Mitchell, "Mitigating unwanted biases with adversarial learning," in *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*. ACM, 2018, pp. 335–340.

[33] H. Edwards and A. Storkey, "Censoring representations with an adversary," *arXiv preprint arXiv:1511.05897*, 2015.

[34] Z. Wu, Z. Wang, Z. Wang, and H. Jin, "Towards privacy-preserving visual recognition via adversarial training: A pilot study," in *Proceedings of the European Conference*

on Computer Vision (ECCV), 2018, pp. 606–624.

[35] C. Xu, J. Ren, D. Zhang, Y. Zhang, Z. Qin, and K. Ren, "Ganobfuscator: Mitigating information leakage under gan via differential privacy," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 9, pp. 2358–2371, 2019.

[36] B. Sadeghi, R. Yu, and V. Boddeti, "On the global optima of kernelized adversarial representation learning," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 7971–7979.

[37] C. Louizos, K. Swersky, Y. Li, M. Welling, and R. Zemel, "The variational fair autoencoder," *arXiv preprint arXiv:1511.00830*, 2015.

[38] R. Zemel, Y. Wu, K. Swersky, T. Pitassi, and C. Dwork, "Learning fair representations," in *International Conference on Machine Learning*, 2013, pp. 325–333.

[39] J. Bringer, H. Chabanne, and A. Patey, "Privacy-preserving biometric identification using secure multiparty computation: An overview and recent trends," *IEEE Signal Processing Magazine*, vol. 30, no. 2, pp. 42–52, 2013.

[40] V. N. Boddeti, "Secure face matching using fully homomorphic encryption," in *2018 IEEE 9th International Conference on Biometrics Theory, Applications and Systems (BTAS)*. IEEE, 2018, pp. 1–10.

[41] L. T. Phong, Y. Aono, T. Hayashi, L. Wang, and S. Moriai, "Privacy-preserving deep learning via additively homomorphic encryption," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 5, pp. 1333–1345, 2018.

[42] S. Avidan and M. Butman, "Blind vision," in *European conference on computer vision*. Springer, 2006, pp. 1–13.

[43] J. Bringer, H. Chabanne, M. Favre, A. Patey, T. Schneider, and M. Zohner, "Gshade: faster privacy-preserving distance computation and biometric identification," in *Proceedings of the 2nd ACM workshop on Information hiding and multimedia security*. ACM, 2014, pp. 187–198.

[44] R. Yonetani, V. Naresh Boddeti, K. M. Kitani, and Y. Sato, "Privacy-preserving visual learning using doubly permuted homomorphic encryption," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2040–2050.

[45] B. Hitaj, G. Ateniese, and F. Perez-Cruz, "Deep models under the gan: information leakage from collaborative deep learning," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2017, pp. 603–618.

[46] Y. Wang, Y.-X. Wang, and A. Singh, "Differentially private subspace clustering," in *Advances in Neural Information Processing Systems*, 2015, pp. 1000–1008.

[47] N. Papernot, M. Abadi, U. Erlingsson, I. Goodfellow, and K. Talwar, "Semi-supervised knowledge transfer for deep learning from private training data," *arXiv preprint arXiv:1610.05755*, 2016.

[48] A. Bhowmick, J. Duchi, J. Freudiger, G. Kapoor, and R. Rogers, "Protection against reconstruction and its applications in private federated learning," *arXiv preprint arXiv:1812.00984*, 2018.

[49] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," *arXiv preprint arXiv:1610.05492*, 2016.

[50] P. Vepakomma, C. Tonde, A. Elgammal *et al.*, "Supervised dimensionality reduction via distance correlation maximization," *Electronic Journal of Statistics*, vol. 12, no. 1, pp. 960–984, 2018.

[51] J. Mairal, "Incremental majorization-minimization optimization with application to large-scale machine learning," *SIAM Journal on Optimization*, vol. 25, no. 2, pp. 829–855, 2015.

[52] ——, "Stochastic majorization-minimization algorithms for large-scale optimization," in *Advances in Neural Information Processing Systems*, 2013, pp. 2283–2291.

[53] M. I. Belghazi, A. Baratin, S. Rajeswar, S. Ozair, Y. Bengio, A. Courville, and R. D. Hjelm, "Mine: mutual information neural estimation," *arXiv preprint arXiv:1801.04062*, 2018.

[54] X. Lin, I. Sur, S. A. Nastase, A. Divakaran, U. Hasson, and M. R. Amer, "Data-efficient mutual information neural estimator," *arXiv preprint arXiv:1905.03319*, 2019.

[55] A. Chaudhuri and W. Hu, "A fast algorithm for computing distance correlation," *Computational Statistics & Data Analysis*, 2019.

[56] X. Huo and G. J. Székely, "Fast computing for distance covariance," *Technometrics*, vol. 58, no. 4, pp. 435–447, 2016.

[57] C. Huang and X. Huo, "A statistically and numerically efficient independence test based on random projections and distance covariance," *arXiv preprint arXiv:1701.06054*, 2017.

[58] C. J. Tonde, "Supervised feature learning via dependency maximization," Ph.D. dissertation, Rutgers University-Graduate School-New Brunswick, 2016.

[59] D. Sejdinovic, A. Gretton, B. K. Sriperumbudur, and K. Fukumizu, "Hypothesis testing using pairwise distances and associated kernels." in *ICML*, 2012.

[60] B. Chang, U. Kruger, R. Kustra, and J. Zhang, "Canonical correlation analysis based on hilbert-schmidt independence criterion and centered kernel target alignment," in *International Conference on Machine Learning*, 2013, pp. 316–324.

[61] R. Poplin, A. V. Varadarajan, K. Blumer, Y. Liu, M. V. McConnell, G. S. Corrado, L. Peng, and D. R. Webster, "Prediction of cardiovascular risk factors from retinal fundus photographs via deep learning," *Nature Biomedical Engineering*, vol. 2, no. 3, 2018.